A Way Out of Dead End Situations in Dialogue Systems for Human-Robot Interaction

Hartwig Holzapfel, Petra Gieselmann

Interactive System Labs,
Universität Karlsruhe,
Am Fasanengarten 5
76131 Karlsruhe, Germany
hartwig@ira.uka.de, petra@ira.uka.de

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

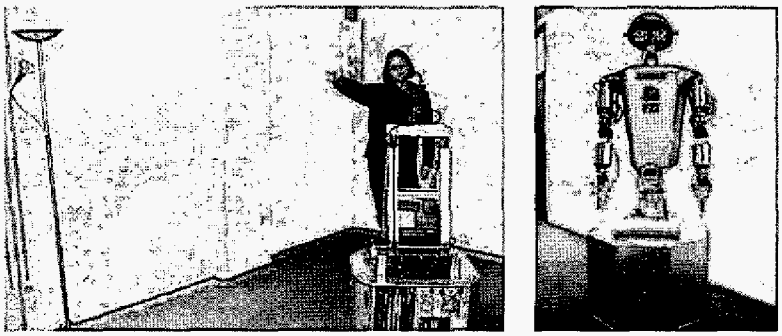
In this paper, we present a strategy for resolving difficult situations in human-robot dialogues where the user input is inconsistent with the current discourse. Reasons for the inconsistency are analyzed in detail and a set of rules is implemented to take all of them into account. In a user test, we evaluated the success of the strategy which can reduce the communication problems resulting from misrecognized user utterances in human-robot communication.

Keywords: human-robot communication; dialogue management

1. Introduction

In recent humanoid robotic systems, speech understanding and natural human-robot interaction have become an interesting research topic [1]. Today, Speech dialogue systems are already commercially available [4], but most of them are very restricted and allow the user only to say some very well defined sentences. They work fine as far as the user says what the system expects. In this paper, we want to explore how a dialogue system can also cope with unexpected situations and how the dialogue can be kept on going in critical situations. This is of special importance in human-robot interaction where everybody should be able to talk to such a robot without any initial training.

Problems occur when the user input is inconsistent with the information already available in discourse. This might result in dead end situations where the user needs a lot of time in correcting the situation. Especially in direct human robot interaction in a household environment, it is important that these situations are avoided so that the user can talk to the robot without any initial training in the same way as with a human servant. Therefore, we evaluated first the preconditions of such a dead end situation and then developed some methods for more efficient dialogue management.



a) b)

Fig. 1. Fig.1 a) Interaction with our development system. Software components include: speech recognition, speech synthesis, person and gesture tracking, dialogue management and multimodal fusion of speech and gestures. Fig 1b): Some components have already been integrated in a humanoid robot with two arms.

Our target scenario is a household situation, in which the user can ask the robot questions related to the kitchen (such as "What's in the fridge ?"), ask the robot to set the table, to switch certain lights on or off, to bring certain objects or to obtain suggested recipes from the robot. The current software components of the robot include a speech recognizer (user-independent large vocabulary continuous speech), a multimodal dialogue component processing speech and gesture input, speech synthesis and the vision-based tracking modules [5]. Pictures of the robotic platform used in the experiments are given in figure 1.

For speech recognition we are using the Ibis decoder [7], which was developed at the University of Karlsruhe as part of our Janus Recognition Toolkit (JRTk) [8]. Besides several other advantages such as smaller memory usage and higher recognition speed, Ibis allows us to decode along context free grammars in addition to the classical statistical n-gram language models and therefore use the same linguistic resources as the dialogue manager. Segmentation is done automatically which leads to recognition errors caused by breathing or laughing for example.

This paper deals with resolving difficult situations in dialogue systems in human-robot communication. Section two gives an overview of related work concerning especially the process of adding new knowledge and integrating it into the discourse. Section three deals with our dialogue manager. Section four gives an overview of our approach to resolve dead end situations. The mechanisms for dialogue state evaluation are also explained. Section five gives experimental details and results, and section five gives a conclusion and outlook.

2. Related Work: Grounding Strategies

Establishing mutual knowledge between the participants in a dialogue is an essential part of the communicative process, in human-human communication as well

as in human-robot communication. This process is called grounding and it concerns adding new information to the common ground of the dialogue participants [10,12,11]. The grounding criterion is reached according to Clark and Schaefer [13], when "the contributor and the partners mutually believe that the partners have understood what the contributor meant to a criterion sufficient for the current purpose". Whereas in human-human communication, we have efficient strategies for managing grounding issues, such as simple feedback strategies, this is a big challenge for spoken dialogue systems which have to deal with imperfect speech recognition. Since cautious grounding with lots of clarification questions from the system leads to very unnatural dialogues, where the user has to confirm everything he said [14], we use more optimistic grounding in our human-robot application. To avoid the obvious drawbacks that the system might work with a wrongly recognized utterance, we implemented a strategy to cope with input inconsistent to the current information in discourse.

Most of the research on the question how an underspecified semantic representation in discourse may be instantiated by new information from the user especially in difficult situations where the input is inconsistent with the discourse information, concerns above all improved clarification questions. Also better integration of the speech recognizer results and especially its confidences into the dialogue system are evaluated in detail. Gabsdil for example uses confidence scores for partial clarification questions in dialogue systems [14]. In this way, the dialogue strategy can be adapted to the contextual plausibility of the speech recognizer's hypotheses. Also Schlangen considers contextual plausibility at various levels of interpretation and uses different kinds of clarification questions depending on that [15].

Since the confidence from the speech recognizer also highly depends on the length of the user utterance, it is hard to compare the confidences for all user utterances in a dialogue. This means that a low confidence from the speech recognizer does not always indicate a wrongly recognized utterance and therefore, we do not want to rely only on this measure. In this paper, our focus is slightly different in using other mechanisms to cope with grounding situations where the information in discourse and the new information from the speech recognizer are inconsistent for some reason. For these cases, we implemented a strategy improving human-robot interaction in real environments.

3. Dialogue Management

For dialogue management we use the TAPAS dialogue tool collection. It is based on the approaches of the language and domain independent dialogue manager ARIADNE [2]. This dialogue manager is specifically tailored for rapid prototyping because only the domain and language dependent components have to be implemented for new applications, whereas the general concepts are already available and can be reused. Furthermore, possibilities to evaluate the dialogue state and general input and output mechanisms are already implemented which can then be applied in the

actual application. For the domain-dependent part, we have developed different kinds of resources: An ontology, a specification of the dialogue goals, a data base, a context-free grammar and generation templates.

The user utterance is parsed by means of a context-free grammar which is enhanced by information from the ontology defining all the objects, tasks and properties about which the user can talk. After parsing, the parse tree is converted into a semantic representation with conversion rules. The semantic representation created during parsing is used to update discourse information. The discourse collects all information that is required to diambiguate the user wish and to reach a dialogue goal. The dialog manager uses type feature structures [9] to represent semantic input and discourse information. If all the necessary information to accomplish a goal is available, the dialogue system calls the corresponding service. But if some information is still missing to accomplish a goal, the dialogue manager generates questions to get this information from the user. This is realized by means of the generation templates which are responsible for generating the spoken output.

3.1. *Abstract Dialogue State*

Many dialogue systems us a compact representation [16] to model the current state of the dialogue. Based on this abstract state the dialogue strategy decides which actions to take. The abstract dialogue state consists of different variables, where each variable describes one aspect of the current state. In previous work, we have suggested variables that model the user's emotional state [5]. The presented dialogue strategy uses additional variables to realize the desired error tolerant behavior. The variables are listed in table 1.

variable	possible values
Intention	selected, determined, finalized, deselected
InputConfidence	float value [0;1]
OverallQuality	good, indermediate1, intermediate2, poor
HoldState	-1, 0, 1

Table 1. Abstract state variables

Some of the variables, Intention, OverallQuality and InputConfidence, have already been used in ARIADNE [3] and proved to be beneficial. The Intention variable describes, how well the discourse information represents the intention of the user, as shown in figure 2. At the very beginning of a dialogue, the Intention value is neutral. During dialogue the system acquires more information that leads to the execution of a dialogue goal. If the collected information is compatible with different dialogue goals, Intention value becomes selected, indicating that the goal of the user still has to be found out. The Intention becomes determined only one goal is selected, then missing information has to be collected that is required to execute the

determined goal. The dialogue goal becomes finalized when all information which is specified in the dialogue goal is available in discourse - this means that all the variables are specified. If the discourse is inconsistent with the dialogue goals the Intention becomes deselected.

The OverallQuality is a quality model and describes the confidence of the system in how well it has been performing during the previous turns. The InputConfidence contains a confidence measure of the current input that is delivered by the multimodal fusion component. The HoldState is a strategy specific variable. It has the value '-1' if the hold strategy is unavailable, '0' if the hodl strategy can be applied, '1' if the hold strategy has been applied for one turn. A detailed description of the strategy can be found in the next section.

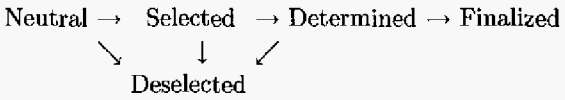


Fig. 2. Intention values and their transitions over different dialogue states

4. Our Approach for Resolving Dead End Situations

Problems occur when the user input is inconsistent with the information already available in discourse. This is the case when the dialogue manager expects an answer to a clarification question as you can see in the example below (see figure 3), but got an unexpected answer. Here the answer 'that one' was wrongly recognized as 'fetch wine' by the system and does therefore not fit to the clarification question from the system.

Therefore, there are two different reasons why the system got an unexpected answer:

(1) There was a speech recognizer error. This means that the user said something different from what has been recognized and therefore, the wrongly recognized utterance cannot be integrated in discourse.
(2) The user did not say what the system expects. For example, the dialogue manager asks a clarification question, but the user does not answer it, but said something else in order to start for example a new subdialogue.

First of all, it is checked whether the user said something correct, but which was misrecognized. Then the possibility that the user wants to start a new subdialogue is evaluated.

User: Switch on that light. + pointing gesture
Recognized Utterance: switch on that light (No gesture recognized)
Robbi: Which lamp do you want me to switch on?
User: Yes Hmm.
Recognized Utterance: yes
Robbi: Which lamp do you want me to switch on?
User: That one.
Recognized Utterance: fetch wine
Robbi: How can I help you?
User: Switch on that light.
Recognized Utterance: switch on that light + pointing gesture to the big lamp
Robbi: Switching on the big lamp.

User: Robbi please switch off the lamp.
Recognized Utterance: robbi please switch off the lamp
Robbi: Going to make you a cup of coffee. Do you want milk?
User: No.
Recognized Utterance: no
Robbi: Do you want sugar?
User: Robbi switch off the lamp.
Recognized Utterance: robbi switch off the lamp
Robbi: Which lamp do you want me to switch off?
User: The big lamp
Recognized Utterance: the big lamp
Robbi: Switching off the big lamp.

Fig. 3. Example Dialogues taken from our user studies with a household robot:1. Misrecognized user utterance 2. Start of a subdialogue

4.1. *Recognition Errors*

There are different criteria which indicate that there might be a recognition error. For example, in this case the confidence for the input hypothesis from the speech recognizer is low or in the n-best list from the speech recognizer a better fitting input hypothesis can be found.

Furthermore, we also check whether the new input can be finalized by itself or whether the old dialogue state was deselected which means that the old input was probably wrong and can therefore be discarded.

In addition, it is checked whether already the last turn contains incompatible information. In this case, the discourse information is discarded. In this way, it can be avoided that the user gets stuck in a non-resolvable situation, but after two turns in the deselected dialogue state, the discourse is cleared and the user can start from

User: Robbi, please set the table.
Recognized Utterance: robbi, please set the table
Robbi: For how many persons do you want me to set the table?
User: Eight.
Recognized Utterance: hello
Robbi: Hello my name is robbi, what can I do for you?
(goal hello finalizes) Robbi: For how many persons do you want me to set the table?
User: For eight persons
Recognized Utterance: for eight persons
...

Fig. 4. stack processing of sub dialogs and dialogue goals

the beginning.

Therefore, we use the following set of rules:

- check confidence of input hypothesis from speech recognizer
- check nbest list for better fitting input hypothesis from speech recognizer
- check if new input can be finalized by itself
- check whether the input was already incompatible with the dialogue state in the last turn

In this way, we can decide whether a new subdialogue should begin or whether the old information should be kept in discourse, although it is not compatible with the new user input. This means that we use a hold strategy for keeping the discourse information in order to be able to deal with recognition errors more efficiently.

4.2. *New Subdialogue*

A new subdialogue is opened, but still the old input is stored. In this way, it is possible to check the next input hypothesis whether it belongs to the subdialogue or not. If so, the subdialogue will be continued, but the discourse state is kept to be able to return to it after the subdialogue will be finished, see figure 4. Otherwise, the subdialogue will be aborted.

4.3. *Decision basis for the dialogue algorithm*

The following variables can be used by the dialogue algorithm (see also table 1):

- Intention variable of the current state
- Intention variable after updating the discourse
- InputConfidence (of speech input)
- n-best list of input hypothesis from speech recognizer
- OverallQuality
- HoldState

The hold strategy is only applied if the current Intention is selected or determined and the previous HoldState is '-1' or '0', then the current HoldState is set to '0'. After conducting the strategy the HoldState is incremented to '1'. If the state is either deselected or finalized, the new discourse contains only the new user input, and the HoldState is set to '-1'. If the previous HoldState is '1' then the hold strategy has already been applied during the previous turn, so the HoldState is set to '-1' which means that for the current turn, the hold strategy is unavailable. The possible transitions of Intention from the current state to the new state are
{"selected","determined"} ->
{"deselected","selected","determined","finalized"}

Updating the discourse includes deciding between the following actions:

- merge discourse information with current user input into one representation
- discard existing discourse information and continue with current user input as discourse information
- open a new subdialogue and put the current discourse on a stack

5. Experimental Details and Results

5.1. *Application Context*

For conducting the experiments, the users interacted with our humanoid household robot [5] which can get cups or dishes, put them somewhere, switch on or off the lights, look in the fridge, give recipe information, etc. The dialogue system can process multimodal commands that combine speech and gesture [5].

Since the system uses automatic segmentation, segmentation is not always correct. The system has to deal with segments that contain noises which are incorrectly interpreted as speech. These errors occur in addition to simple recognition errors, as you can see in table 2.

Currently we are using only close talking microphones for speech input. Incorrect segmentation is mostly only lip smacks or breathing. In the future we want to start using distant speaking microphones, and thus it will become even more important to handle incorrect input with algorithms like the presented one.

5.2. *Collected Data*

The users were asked to instruct the robot to set the table. The operation / goal 'setting the table' requires the parameters (i) for how many people, (ii) what kind of glasses are put on the table and (iii) if dessert will be served.

We applied two dialogue strategies to the user input that were compared to each other afterwards. The baseline dialogue strategy follows each user input and creates either a new subdialogue or discards the old discourse state and continues with the new state. The second dialogue strategy applies the hold strategy for one turn as already described in section 4.

To test the behaviour in incorrect states and the impact of incorrect holds, we added a component to simulate recognition errors. This component has a predefined probability by which it substitutes the speech recognizer's output. This is only done while no dialogue goal is selected and leads to the selection of a wrong dialogue goal. This way we can test how the hold strategy influences (negatively) wrong decisions of the system. We have chosen a probability of 50%. For example the user says 'please set the table' which is correctly recognized by the system, but replaced by 'please make me a coffee'. The system then responds 'I am going to make you a coffee. Do you want milk?'. The system is then in the state 'determined' where it tries to acquire information to finalize the determined goal. Since this is not the goal that the user has, the user has to tell the system to abandon the current goal and select the correct goal.

The optimal strategy to recover from this error state would be to utter 'start over'. This is a technical command that resets the current dialogue state. The tested strategy also allows to execute any input that directly leads to a finalized goal, which in this case would reset the discourse, e.g. 'hello'.

Simply speaking 'please set the table' would not result in the desired behaviour, since the system follows the hold strategy. We classify this as an incorrect hold decision. However, this didn't seem to be a problem, since the users spoke the same sentence again, which would then lead to the correct goal (algorithm see section 4). Furthermore, some users also started to use the optimal strategy after some time, while getting more familiar with the system.

5.3. *Evaluation*

We performed a small evaluation with four speakers, applying the hold strategy. (We are currently recording more user tests, the result will be integrated into the camera ready version of this paper).

We found several examples of the successful hold strategy as you can see in figure 5 where the wrongly recognized utterance 'put forks please there' does not result in aborting the whole dialogue goal, but the discourse information is kept so that the next user utterance which is recognized correctly leads to finalizing the dialogue goal successfully.

The overall completion rate of the dialogue system is evaluated by means of counting the number of times a dialogue goal is aborted, respectively successfully finalized. The hold strategy could be evaluated directly from the recorded data.

5.4. *Results*

As you can see in the table below (see table 3), the number of dialogue goals which can be finalized increased substantially by means of the hold strategy.

Table 4 shows detailed information about each user's interactions with the system.

User: Robbi, please set the table.
Recognized Utterance: robbi, please set the table
Robbi: For how many persons do you want me to set the table?
User: Eight persons.
Recognized Utterance: take espresso
Robbi: For how many persons do you want me to set the table?
User: For eight persons.
Recognized Utterance: for eight persons
Robbi: Do you want long drink glasses or wine glasses?
User: Long drink glasses.
Recognized Utterance:rec: long drink glasses
Robbi: Do you want your meal with or without dessert?
User: Of course with dessert.
Recognized Utterance: put forks please there
Robbi: Do you want your meal with or without dessert?
User: With dessert.
Recognized Utterance: with dessert
(goal finalized)

Fig. 5. Example of a successful dialog with hold strategy

User: Robbi, please set the table.
Recognized Utterance: robbi, please set the table
Robbi: For how many persons do you want me to set the table?
User: Eight persons.
Recognized Utterance: take espresso
Robbi: How can I help you?
(goal aborted)

Fig. 6. Example of a successful dialog without hold strategy

	segmentation errors	sentence error rate
speaker 1 (in %)	2	39
speaker 2 (in %)	23	47

Table 2. sentence error rate, including segmentation errors (full evaluation has not ready for the submission, it will be inserted for the final version)

6. Conclusions and Outlook

In this paper, we presented a new dialogue strategy that improves human-robot interaction in real environments. The so called 'hold strategy' keeps the discourse information although it is inconsistent with the new user utterance and therefore

	Aborted Goals	Finalized Goals
Strategy without Hold (in %)	84.62	15.38
Strategy with Hold (in %)	69.23	30.77

Table 3. Overall rate of aborted, respectively finalized goals with the hold strategy

user 1	7 started goals
default strategy	1 goal finalized
hold strategy	4 goals finalized 4 correct holds, 3 incorrect holds
user 2	10 started goals
default strategy	0 goals finalized
hold strategy	2 goals finalized 8 correct holds, 3 incorrect holds
user 3	8 started goals
default strategy	4 goals finalized
hold strategy	4 goals finalized 6 correct holds, 5 incorrect holds
user 4	14 started goals
default strategy	1 goal finalized
hold strategy	2 goals finalized 7 correct holds, 8 incorrect holds

Table 4. Overall rate of aborted, respectively finalized goals with the hold strategy

reduces the problems resulting from misrecognized utterances. In this way, a single wrongly recognized utterance does not lead to aborting the dialogue goal, but the user can still go on with a started dialogue goal.

In a user study, we tested this hold strategy. The results are promising showing that about 31% of the dialogue goals can be finalized with the hold strategy compared to 15% without the strategy.

In the future, we want to improve this strategy by adding more specific clarification strategies and different kinds of clarification requests given the fact that the results of other researchers in this field seem to be very promising [14,15].

Furthermore, we will extend the use of n-best lists from the speech recognizer so that the dialogue manager evaluate the n-best list in the current discourse context in order to decide then which is the best hypothesis given the dialogue context.

Acknowledgements

This work was supported in part by the German Research Foundation (DFG) as part of the SFB 588 and within the FAME project by the European Union as

IST-2000-28323 .

References

1. Johan Bos and Ewan Klein and Tetsushi Oka, Meaningful Conversation with a Mobile Robot, *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Hungary, 2003.
2. Matthias Denecke, Rapid Prototyping for Spoken Dialogue Systems, *Proceedings of the 19th International Conference on Computational Linguistics*, Taiwan, 2002.
3. Matthias Denecke, Informational Characterization of Dialogue States, *Proceedings of the International Conference on Speech and Language Processing*, Beijing, China, 2000.
4. Michael F. McTear, Spoken Dialogue Technology: Enabling the Conversational Interface, *ACM Computing Surveys*, volume 34(1), pp. 90–169, 2000.
5. P. Gieselmann and C. Fuegen and H. Holzapfel and T. Schaaf and A. Waibel, Towards Multimodal Communication with a Household Robot, *Third IEEE International Conference on Humanoid Robots (Humanoids)*, Karlsruhe, Munich, Germany, 2003.
6. R. Stiefelhagen and C. Fuegen and P. Gieselmann and H. Holzapfel and K. Nickel and A. Waibel, Natural Human-Robot Interaction using Speech, Gaze and Gestures, *submitted to IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.
7. H. Soltau, F. Metze, C. Fügen and A. Waibel, A One pass- Decoder based on Polymorphic Linguistic Context Assignment, *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU-2001)*, 2001.
8. M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries and M. Westphal, The Karlsruhe-Verbmobil Speech Recognition Engine, *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP-97)*, 1997.
9. Bob Carpenter, The Logic of Typed Feature Structures, Cambridge University Press, 1992.
10. David R. Traum, Computational Models of Grounding in Collaborative Systems, *Psychological Models of Communication in Collaborative Systems - Papers from the AAAI Fall Symposium*, 1999.
11. Massimo Poesio and David Traum, Towards an Axiomatization of Dialogue Acts, J. Hulstijn and A. Nijholt (eds.), Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology), Enschede, pp., 207–222, May 1998.
12. David R. Traum and Pierre Dillenbourg, Towards a Normative Model of Grounding in Collaboration, *Working notes of the ESSLLI-98 workshop on Mutual Knowledge, Common Ground and Public Information*, 1998.
13. H. H. Clark and E. F. Schaefer, Contributing to discourse, *Cognitive Science* 13:259-294, 1989
14. Malte Gabsdil, Clarification in Spoken Dialogue Systems, *Proceedings of the 2003 AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue*, pp. 28–35, Stanford, CA, 2003.
15. David Schlangen, Causes and Strategies for Requesting Clarification in Dialogue, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue (SIGDIAL 04)*, Boston, USA, 2004.
16. Marilyn Walker, An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email, *Journal of Artificial Intelligence Research*, Vol 12., pp. 387–416, 2000.